



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/736,324	12/15/2000	Bela Bodo	040010-907	8709

27045 7590 12/31/2003

ERICSSON INC.
6300 LEGACY DRIVE
M/S EVW2-C-2
PLANO, TX 75024

EXAMINER

VU, TUAN A

ART UNIT	PAPER NUMBER
----------	--------------

2124

DATE MAILED: 12/31/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/736,324

Applicant(s)

BODO, BELA

Examiner

Tuan A Vu

Art Unit

2124

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 15 December 2000.
- 2a) ☐ This action is FINAL. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-6 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-6 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 15 December 2000 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. §§ 119 and 120

- 12) ☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
a) ☒ All b) ☐ Some * c) ☐ None of:
1. ☒ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
* See the attached detailed Office action for a list of the certified copies not received.
- 13) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application) since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.
a) ☐ The translation of the foreign language provisional application has been received.
- 14) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121 since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) 2
- 4) ☐ Interview Summary (PTO-413) Paper No(s) _____
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: _____

3333DETAILED ACTION

1. This action is responsive to the application filed December 15, 2000.

Claims 1-6 have been submitted for examination.

Priority

2. Acknowledgment is made of applicant's claim for foreign priority based on an application filed in Sweden on 12/17/1999. Receipt is acknowledged of papers submitted under 35 U.S.C. 119(a)-(d), which papers have been placed of record in the file.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

4. Claims 1-2 are rejected under 35 U.S.C. 103(a) as being unpatentable over McLain, Jr, USPN: 5,956,513 (hereinafter McLain), in view of Leblang et al., USPN: 5,574,898 (hereinafter Leblang); and further in view of Hiller et al., USPN: 6,658,659 (hereinafter Hiller)

As per claim 1, McLain discloses a method for tracing the errors in executable software of a computer controlled system, said software is compiled and linked in a building process, wherein a number of source-code files stored in a version control system (e.g. col. 3, line 65 to col. 4, line 16 – Note: read configuration file to determine version correspondence is implicitly disclosing version control system), and part of said building process further results in a record (*configuration data file, internal table* – col. 4, line 4-46) which specifies names and versions of source-code files included in said build process, such process including the steps:

Art Unit: 2124

storing said record in a version controlling system (e.g. col. 13, line 52 to col. 14, line 14; Fig. 5,6a-e – Note: configuration file linking to header files with version information discloses version controlling system for administrating a record),

retrieving path and version number of the header or library files specified by the record (e.g. *latest version, path used*– Fig. 6a-e; col. 15, lines 3-43); and

bundling said path and version number with said executable software(e.g. col. 11, lines 6-27).

But McLain does not explicitly specify retrieving the version number and path is actually retrieving path and version of the record itself. But McLain discloses a record being specific to a build and a version control associated with a build control, hence suggest a identification number associated with a build (e.g. *ABC 125* - Fig. 1). Analogous to the concept of providing a specific identification to a release or a build thus as suggested by McLain, Leblang, in a method to use version control within a software build, discloses a record or configuration data pointing to versioned objects or header files (e.g. Fig. 23) similar to McLain's, and further associates a version number and location to a global tree representing a release for check-in/check-out (e.g. *version of directory include* - Fig. 17-18; col. 28, lines 21-47); and encompasses build information and execution instruction in a script (e.g. *clearmake* - col. 28, line 48 to col. 29, line 43). It would have been obvious for one of ordinary skill in the art at the time the invention was made to provide path and version information to the build record as suggested by Leblang in using directory version and build script, each of which entails path and version associated with a build or release as suggested above, to the build method of McLain. One of ordinary skill in the art would be motivated to do so in such case because providing the administrating of versioned

Art Unit: 2124

high level record (e.g. versioned directory) and build script as taught by Leblang would enhance the view of a particular build among other builds within a hierarchy of build legacy and further would enable the use of one file, e.g. script to encompass the instructions and retrieving information on versioned elements or files to be included in the build, thus achieve version checking at a higher level than file level granularity and facilitating speedy auditing of a build (Leblang: col. 30, line 47 to col. 31, line 29).

Nor does McLain specify that bundling record path and version number is in such a way that said path and version number is retrievable at the site where the executable is to be used. In addition to the use of script to encompass versioned objects and instructions purported to effect version auditing and a build of a specific version/release Leblang discloses a transparent file distribution system via a network of multiple developers communicating with private workstations (Fig. 2, 7) and use of make file (col. 14, line 49 to col. 16, line 16). Additionally, including a path and version number in Unix MAKE file is a known concept at the time the invention was made. In the same line of software verification such as to audit software or perform compatibility check by a processing system prior to a build analogous to that of McLain and Leblang, Hiller discloses receiving a software via network interface (col. 13, lines 24-54) and using of loading module and identify header information for version compatibility and directory path check (e.g. col. 4, lines 37-56; col. 6, line 27 to col. 7, line 5; Fig. 2B) analogous to the error checking system by McLain or the auditing script by Leblang. It would have been obvious for one of ordinary skill in the art at the time the invention was made to provide distributing of error checking software to remote station as taught by Hiller and bundling of path and version of record or build script as taught by Leblang to McLain's method of building, in

Art Unit: 2124

case such record is used to provide the build or to activate the received software in a remote workstation as taught by Hiller. The motivation is to provide sufficient information for the receiving workstation to retrieve the script or build record just so to effect the build and enforce the correctness checking as taught by Leblang, because this would alleviate further the resources usage at the executing remote workstation since limit in resources might be an issue many of these network devices.

As per claim 2, McLain does not specify post-processing of the executable file with integrating of path and version number therein. But the limitation to provide path and version identifying the record or the script aimed at effecting the build and the version compatibility check has been addressed in claim 1 above using Leblang's teachings and Hiller's loading of software; and the corresponding rejection as set forth in claim 1 herein applies for the same motivation.

5. Claim 3 is rejected under 35 U.S.C. 103(a) as being unpatentable over McLain, Jr, USPN: 5,956,513; Leblang et al., USPN: 5,574,898; and Hiller et al., USPN: 6,658,659; as applied to claim 1 above; and further in view of Thomas et al., USPN: 6,460,052 (hereinafter Thomas).

As per claim 3, McLain does not specify storing the executable in a PROM. However, Leblang suggests network multi-station communicating of auditing script and Hiller teaches about network receiving and loading of executable with version/path checking (re claim 1). In small devices known to be used a processing system in the internet, the storage resources therein being a limiting factor was a known concept at the time of the invention. As an evidence to that fact, Thomas discloses storing of versioned files in a multi-developer similar to that of Leblang,

Art Unit: 2124

and further discloses storing of code in PROM within a wireless network communication system (e.g. col. 15, line 41 to col. 16, line 14). It would have been obvious for one of ordinary skill in the art at the time the invention was made to provide distributing of software as suggested by Hiller and Leblang for providing executable at the receiving machine/device so that the executable can be stored in a PROM as suggested by Thomas because the programmable memory unit would accommodate for the lack of storage resources so well-known in small devices such as observed above and illustrated via the wireless communication as suggested by Thomas.

6. Claims 4-6 are rejected under 35 U.S.C. 103(a) as being unpatentable over McLain, Jr, USPN: 5,956,513; Leblang et al., USPN: 5,574,898; and Hiller et al., USPN: 6,658,659; and further in view of Hammond, USPN: 5, 974,470 (hereinafter Hammond).

As per claim 4, McLain discloses a method for tracing the errors in executable software of a computer controlled system, said software is compiled and linked in a building process, wherein a number of source-code files stored in a version control system (e.g. col. 3, line 65 to col. 4, line 16 – Note: read configuration file to determine version correspondence is implicitly disclosing version control system), and part of said building process further results in a record (*configuration data file, internal table* – col. 4, line 4-46) which specifies names and versions of source-code files included in said build process, such process including the steps: compiling source code into object files and linking said files (Fig. 2, 3A), wherein said compiling and linking steps also creates a record specifying names and versions of used source code files (*configuration data file, internal table* – col. 4, line 4-46);

Art Unit: 2124

storing said record in a version controlling system (e.g. col. 13, line 52 to col. 14, line 14; Fig. 5, 6a-e – Note: configuration file linking to header files with version information discloses version controlling system for administrating a record),

retrieving path and version number of the header or library files specified by the record (e.g. *latest version, path used*– Fig. 6a-e; col. 15, lines 3-43); and

creating an object file with including said path and version number (e.g. col. 11, lines 6-27).

But McLain does not specify linking the source files into a relocatable module. But in view of McLain's teaching about locating of source files or header file during the linking process using a configuration file (e.g. Fig. 2, 3A, 6A-D), the limitation as to link files into a relocatable module is implicitly disclosed because if a linked file is not relocatable subsequent linking using such file would not be able to achieve the generation of the object code.

Nor does McLain disclose retrieving path and version of the record thus stored. But this limitation has been addressed in claim 1 above using Leblang's teachings.

Nor does McLain disclose creating and compiling a source code file where path and version number are defined as global variables but McLain discloses using of a global record to store information as to linking modules. The limitation to incorporate path and version of the record in the final executable code has been addressed in claim 1 above. Further, in the context as to support the executing environment that receives the software transferred for activation therein as suggested by Hiller, Hammond, in a system to load dynamic linked libraries in a window machine using version compatibility checking analogous to McLain's error checking method, discloses global variables stored in a database to enforce rules for building the module

Art Unit: 2124

and invoking the functions for activating the dynamic libraries, wherein such global variables include path information of functions needed for the build (e.g. col. 14, line 24 to col. 15, line 29). In view of the teachings by Leblang to include path and version information of the auditing script to the version compatibility and build method by McLain, it would have been obvious for one of ordinary skill in the art at the time the invention was made to implement those path and version data as global variables imparted to the global record as taught by McLain or to the script by Leblang (re claim 1) or to the source code to be built or compile such as suggested by Hammond from above because these path and version data would be the uppermost piece of information to parse in the source code in order to have an immediate and starting point to retrieve instructions to retrieve linking files and libraries as intended by McLain or Leblang.

As per claim 5, McLain discloses a method for tracing the errors in executable software of a computer controlled system, such method includes the step of :

storing (said record);

retrieving (path and version number); and

bundling (path and version) as recited in claim 1 above.

These step limitations have been recited in claim 1 and are now rejected using the same rationale as set forth in claim 1 above.

But McLain does not specify that the executable software is function library software. Hammond, in a similar method to provide a build with version checking for modules to assemble and activate in the process of evoking a software application, disclose building and activating a dynamic linked library (e.g. Fig. 1-4). It would have been obvious for one of ordinary skill in the art at the time the invention was made to implement the executable as suggested by McLain so

Art Unit: 2124

that it is a function library software as suggested by Hammond, because the library software is a very portable and handy software which does not require excessive storage resources nor does it require re-compiling utilities in the environment that stores it for the intent to activate it.

As per claim 6, the combination of McLain and Leblang discloses including path and version number in the record used to configure variables and set up instructions in order to effect the version compatibility check and build as mentioned and addressed in claim 1. But neither McLain nor Leblang disclose defining those path and version data as a global string variables such that they are formatted like “@(#) path” or “@(#) version number”. The limitation as to provide path and version as global variables has been addressed above using Hammond’s teachings. Further, official notice is taken that the use of “#” symbol in macro or directive such as in C/C++ compiler for defining global macro variables was a well known concept in the art of programming at the time of the invention. Hence providing a “@” in front of a “#” to distinguish a macro or global parameters as used in the build combination of McLain/Leblang/Hiller/Hammond would have been but one slight variance of the well-known format used in the well-known programming language. Thus, it would have been obvious for one of ordinary skill in the art at the time the invention was made to provide a “@(#) Path/Version” as format for a global variable for the auditing and build scheme by the combination of McLain/Leblang/Hiller/Hammond as set forth in claim 5 above. The motivation is that a variance from a standard “# global var” would enable the global variable such as defined above to be differentiate from the standard variables used for the execution of the code in that the global variables such as path and version are exclusively to provide where to obtain the record

Art Unit: 2124

which actually contains instructions and relocatable data for linking of the actual software to build and execute.

Conclusion

7. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U.S. Pat No. 5,748,961 to Hanna et al., disclosing path name in Unix makefile for build process.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (703)305-7207. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (703)305-9662.

Any response to this action should be mailed to:

Commissioner of Patents and Trademarks

Washington, D.C. 20231

or faxed to:

(703) 872-9306 (for formal communications intended for entry)

or: (703) 746-8734 (for informal or draft communications, please label

“PROPOSED” or “DRAFT” – Please consult Examiner before use)

Hand-delivered responses should be brought to Crystal Park II, 2121 Crystal Drive, Arlington. VA. , 22202. 4th Floor(Receptionist).

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Application/Control Number: 09/736,324

Page 11

Art Unit: 2124

Kakali Chaki

VAT

December 26, 2003

**KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100**